Reduction Rules for Interaction Graphs

Daniela Weinberg

Humboldt–Universität zu Berlin
Institut für Informatik
D–10099 Berlin
weinberg@informatik.hu-berlin.de

Abstract. The internet today has grown to be more than just being a basis for exchanging information. It steadily becomes a platform for processing business processes. Many companies distribute their service with the help of web services or integrate other web services into their own workflow.
However, before a web service gets published it should be examined well. We will introduce a way of examining the *controllability* of a web service. That means, we study whether a controller can actually use the functionality provided by the web service. We propose the *interaction graph* of a web service, that is modelled by an open workflow net. To verify whether such a net is controllable or not it is sufficient to construct a reduced interaction graph. We will define reduction rules that minimize the size of the graph greatly. The analysis using the interaction graph as well as the reduction rules shown in this paper are implemented and have been integrated into an analysis tool kit for web services.

1 Introduction

In these days enterprises tend to source out functionalities and cooperation across borders has become increasingly important. For specific tasks so-called virtual enterprises are being formed. In this setting, *services* play an important role. Such a service basically encapsulates self-contained functions that interact through a well-defined interface. We assume the essentials of a service to include an *identifier* (id), its *interface*, and its *operational behavior*. With the help of the interface the service can communicate with its environment during the execution. The operational behavior of a service is a set of operations to be executed according to some internal control structure. The well-known class of *web services* is an implementation of services with an interface specified in WSDL [AIM01] and an id given by an URI. Throughout this paper we focus on services with operational behavior that is described as a *workflow*. We call such a service *workflow service*. With the raise of the language BPEL [BIMS02] the class of workflow services has become more and more important. BPEL provides a means to describe workflow behavior using certain control structures. It is a notation for web services whose control structure is modelled as a workflow.

A common example of a workflow service is a travel agency, that usually combines several web services. Surely a travel agency might also be a web service. A Java program, for instance, is definitely no workflow service, but it might implement a web service.

Before deploying a workflow service it is of great importance to analyze it thoroughly. A workflow service provides certain functionality. Therefore it is advisable to analyze whether this functionality can be used by another service. That means whether it is *controllable* or not. In this paper we consider workflow services as nets having an interface – the *open workflow nets*.

For those nets a technique for analyzing the controllability has been introduced in [Mar03a] – the *communication graph*. The edges of those graphs represent communication steps that stand for consuming and producing messages by the net. The nodes of the graph are divided into visible nodes indicating the start and the end of each step and not visible nodes depicting the state of the net between an input and an output of the controller. The graph therefore is compressed into communication steps. The calculation of each communication step has some implicitly defined reduction rules. In order to analyze whether the net is controllable the complete graph has to be calculated.

In this paper we will introduce a different technique for examining the controllability of open workflow nets. That is, we will define the *interaction graph* for an open workflow net which let's us decide whether the net is controllable or not. The complete interaction graph, however, shows all the communication that is possible between the net and its controller. It represents all reachable states of the net being analyzed while communicating with its controller. Therefore the complete graph is huge in size (comparable to the reachability graph of Petri nets). However, this way it is possible to apply specific reduction rules while building up a reduced graph. As our case studies show, the resulting graph is extremely smaller then the complete graph. With the help of our case studies we could even show that the reduced interaction graph usually is smaller than the communication graph for the corresponding net.

The rest of this paper is divided into the following sections. Section 2 provides the basics of modeling workflow nets. The next section 3 introduces the interaction graphs and defines the control strategy as part of the graph. Furthermore it motivates the next section by speaking about the complexity of interaction graphs. Section 4 is the main part of this paper. Here we define different reduction rules that can be used to reduce the size of the interaction graph with respect to the controllability analysis. Section 5 shows the results of our case studies we have done to test our approach in practice. In the last two sections 6 and 7 we give a summary of this paper and speak about our future work.

2 Open Workflow Nets

A model for workflows has already been suggested by van der Aalst [vdA98]. He defines a special class of Petri nets, the so-called *workflow nets* (WFNs), that describe the control structure of workflows in an adequate way. Since workflow services are supposed to communicate with other workflow services, additional constructs for modeling communication channels are needed. We use *open workflow nets* (oWFNs) that were introduced in [MRS05]. Those nets are essentially a liberal version of van der Aalst workflow nets, enriched with communication places. Each communication place of an oWFN models a channel to send (receive) messages to (from) another oWFN. This way we abstract from data and model the occurance of messages just as undistinguishable tokens.

We assume the usual representation of Petri nets $N = (P, T, F)$, with P being the set of *places* (graphically, circles) and T being the set of *transitions* (graphically, squares). The set $F \subseteq (P \times T) \cup (T \times P)$ represents the arcs of the net that are depicted as arrows. A marking is a mapping $m : P \to \mathbb{N}$ (graphically, $m(p)$ black tokens on place p). A transition t is *enabled* at a marking m if for each place p with $(p, t) \in F, m(p) \geq 1$.

If enabled at m, the *occurance* of t then yields the marking m' with $m'(p) = m(p) - 1$ if $(p, t) \in F$ and $(t, p) \notin F, m'(p) = m(p) + 1$ if $(t, p) \in F)$ and $(p, t) \notin F$, and $m' = m(p)$ otherwise.

Definition 1 (Open Workflow Net)
An *open workflow net* is a Petri net $N = (P, T, F)$ together with

1. two sets $in, out \subseteq P$, such that for all transitions $t \in T$ holds: if $p \in in$ ($p \in out$) then $(t, p) \notin F$ ($(p, t) \notin F$),
2. a distinguished marking m_0, called the *initial marking*, and
3. a set Ω of distinguished markings, called the *final markings* of N. $\star$

The places in *in* (*out*) are called *input* (*output*) places. The set $in \cup out$ is called the *interface* of N. The *inner* of N can be obtained from N by removing all interface places, together with their adjacent arcs. We label a transition t connected to an input (output) place x with $?x$ and name it *receiving transition* ($!x$, *sending transition*). A transition that is not connected to an interface place is called τ transition.

The interaction of two oWFNs is reflected by their *composition*. We assume that the oWFNs M and N share input- and output elements: $(P_M \cup T_M) \cap (P_N \cup T_N) \subseteq (in_M \cup out_M) \cap (in_N \cup out_N)$. The composition of M and N yields a new oWFN, denoted by $M \oplus N$. It is constructed by the component-wise union of M and N. Let $M \oplus N$ be defined by $P_{M \oplus N} =_{def} P_M \cup P_N$, $T_{M \oplus N} =_{def} T_M \cup T_N$, $F_{M \oplus N} =_{def} F_M \cup F_N$. Each place in $out_M \cap in_N$ (or in $in_M \cap out_N$) turns into an inner place of $M \oplus N$. With $I =_{def} (out_M \cap in_N) \cup (in_N \cap out_M)$, let $in_{M \oplus N} =_{def} (in_M \cup in_N) \setminus I$ and $out_{M \oplus N} =_{def} (out_M \cup out_N) \setminus I$. For markings m_M of M and m_N of N let $m_M \oplus m_N$ be a marking of $M \oplus N$, defined for $p \in P_{M \oplus N}$ by $(m_M \oplus m_N)(p) =_{def} m_M(p) + m_N(p)$, where $m_M(p) = 0$ if $p \notin P_M$ and $m_N(p) = 0$ if $p \notin P_N$. Then, let $m_{(M \oplus N)_0} =_{def} m_{M_0} \oplus m_{N_0}$ and $m_{M \oplus N} \in \Omega_{M \oplus N}$ iff $m_{M \oplus N} = m_M \oplus m_N$ for some $m_M \in \Omega_M$ and some $m_N \in \Omega_N$.

A marking m of a oWFN is a *deadlock* if m enables no transition at all. An oWFN in which all deadlocks are final markings is called *weakly terminating*. Given an oWFN N, we call an oWFN M a *strategy* for N iff the oWFN $N \oplus M$ is weakly terminating. N and M then are *partners*.

Definition 2 (Controllability)
Let N be an oWFN. N is controllable, if there exists an oWFN M, such that the composed oWFN $N \oplus M$ weakly terminates. $\star$

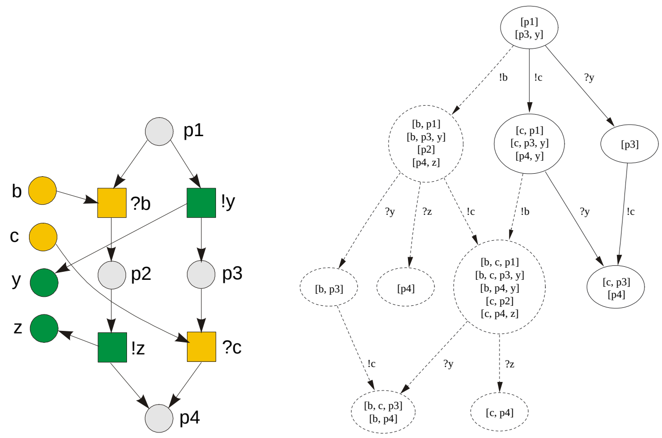
Throughout this paper we refer to M as a *controller* of N and we call a marking a *state* of the net. Further we only consider acyclic open workflow nets and we just permit those final markings that have empty interface places. It is part of further research to adapt the results shown in this paper to those nets having final markings that do not necessarily have empty interface places.

3 Interaction Graphs

The *interaction graph* (IG) of an oWFN has been developed with the reachability graphs of Petri nets [CST02,Sta90] in mind. In contrast to those graphs it represents the controller's point of view. The nodes of the graph are a set of states, which the net can reach by consuming and producing messages. The edges of the graph represent the actions of the controller – sending and receiving messages. Basically, the nodes of the graph are a *hypothesis* of the controller with respect to the state of the net. The controller only knows in which set of states the net is in. It does, however, not know the exact state of the net.

There is communication between the controller and the net. The controller can control the net in a limited way by sending messages. Whereas by receiving messages from the net, the controller gets some knowledge about the state the net might be in. We distinguish two kind of events: (1) *sending event* means that the controller sends a message to the net (labelled by !) and (2) *receiving event* represents the receiving of a message (labelled by ?) by the controller.

Example 1. Figure 1 shows the oWFN N1 (Fig. 1(a)) and its interaction graph IG(N1) (Fig. 1(b)). N1 possesses the interface places $\{b, c\} = in(N1)$ and $\{y, z\} =$



(a) N1 (b) IG(N1)

Fig. 1. oWFN N1 and its interaction graph

$out(N1)$. Further, $inner(N1) = \{p1, p2, p3, p4, ?b, !y, !z, ?c\}$. The initial marking of N1 is $[p1]$ and the final marking is $[p4]$. ⋆

A net can change its state on its own or by messages being received from the controller. We differentiate two kinds of states a net can be in. Based on that we define a set of states a net can be in without having to interact with the controller.

Definition 3 (State Set of an oWFN)
Let $N = (P, T, F)$ be an oWFN and let z be a state of N. We call state z (i) *maximal*, if $\{z' \mid \exists t \in T : z \xrightarrow{t} z'\} = \emptyset$ and (ii) *transient*, otherwise. The set $\mathcal{Z}$ is a *state set of* N, iff $\forall z \in \mathcal{Z} : (z \text{ is transient} \wedge \exists t \in T\, \exists z' \in \mathcal{Z} : z \xrightarrow{t} z') \vee (z \text{ is maximal})$. ⋆

The root node of the IG in Fig. 1(b) contains the state set $\{[p1], [p3, y]\}$. The state $[p1]$ is transient, since the net can switch to state $[p3, y]$ on its own. This state is maximal. N1 cannot switch to another state on its own. Therefore the two states $[p1]$ and $[p3, y]$ make up a state set.

A state of an oWFN is a multiset. The sum of two multisets $P+Q$ is the multiset where the multiplicity of an element in $P+Q$ is equal to the sum of the multiplicities of the element in P and in Q.

Each state set of an oWFN can activate sending- and receiving events. We use the following definition for computing the sets of events that are activated within a state set.

Definition 4 (Activation of Sending- and Receiving Events)
Let $\mathcal{Z}$ be a state set of an oWFN N. The sending event $!A = \{a_1, \ldots, a_n\}$ with $A \subseteq in(N)$ is *activated in* $\mathcal{Z}$, iff $\exists z \in \mathcal{Z} : z + a_1 + \cdots + a_n \xrightarrow{?a_1,\ldots,a_n} z' \wedge z[a_1] = \cdots = z[a_n] = 0$. The receiving event $?B = \{b_1, \ldots, b_m\}$ *is activated in* $\mathcal{Z}$, iff $\exists z \in \mathcal{Z} : z[b_1] \geq 1 \wedge \cdots \wedge z[b_m] \geq 1 \wedge B \subseteq out(N)$. ⋆

Definition 5 (Activated Sending- and Receiving Events)
Let $\mathcal{Z} = \{z_1, z_2, \ldots, z_n\}$ be a state set of an oWFN N, let $i \in \{1, \ldots, n\}$, let $!A$ be a sending event and $?B$ a receiving event. The following sets are defined for $\mathcal{Z}$.

– *Set of activated sending events*: $\mathcal{S}(z_i) = \{A \mid \exists z_i' : z_i + A \xrightarrow{?A} z_i' \wedge z_i[A] = 0 \wedge A \subseteq in(N)\}$ and $\mathcal{S}(\mathcal{Z}) = \mathcal{S}(z_1) \cup \mathcal{S}(z_2) \cup \cdots \cup \mathcal{S}(z_n)$
– *Set of activated receiving events*: $\mathcal{R}(z_i) = \{B \mid z_i[B] \geq 1 \wedge B \subseteq out(N)\} \cap \{\{p\} \mid p \in out(N)\}$ and $\mathcal{R}(\mathcal{Z}) = \mathcal{R}(z_1) \cup \mathcal{R}(z_2) \cup \cdots \cup \mathcal{R}(z_n)$ ⋆

Lets take a look at Fig. 1(b) again. The root node activates the sending events $!b$ and $!c$, because $[p1] + [b] \xrightarrow{?b} [p2]$ and $[p3, y] + [c] \xrightarrow{?c} [p4]$ with $b, c \subseteq in(N1)$. That means, the sum of two multisets leads to a new multiset, which is a new state of the net. The node further activates the receiving event $?y$, because $[p3, y][y] = 1$ (multiplicity of y in the multiset $[p3, y]$) and $y \subseteq out(N1)$. So, we have for the root node $\mathcal{A}(N1) = \{[p1], [p3, y]\}$: $\mathcal{S}(\mathcal{A}(N1)) = \{b, c\}$ and $\mathcal{R}(\mathcal{A}(N1)) = \{y\}$.

In order to compute the interaction graph we need to calculate the successor state set based on the actual state set.

Definition 6 (Computation of the Successor State Set)
Let $\mathcal{Z}$ be a state set of an oWFN N. Let $!A$ be a sending event with $A = \{a_1, \ldots, a_n\} \in \mathcal{S}(\mathcal{Z})$. Let $?B$ be a receiving event with $B = \{b_1, \ldots, b_m\} \in \mathcal{R}(\mathcal{Z})$. The successor state set $\mathcal{Z}'$ is computed with respect to the type of event (sending, receiving):

(a) *sending event*: $\mathcal{Z}' = \{z' \mid z \in \mathcal{Z} : z + a_1 + \cdots + a_n \xrightarrow{*} z'\}$
(b) *receiving event*: $\mathcal{Z}' = \{z - b_1 - \cdots - b_m \mid z \in \mathcal{Z} \wedge z[b_1] \geq 1 \wedge \cdots \wedge z[b_m] \geq 1 \wedge B \subseteq out(N)\}$ ⋆

The successor state set of a sending event $!A$ contains all states, that can be reached from the actual state set by sending the messages A: (1) The messages A might remain in the message channels. Thus, the states $\mathcal{Z} + A$ are added to the successor state set. (2) All those states are added to the successor state set, that can be reached by firing transitions. So, we calculate the successor states of the transient states, that we just added to the successor state set. In particular the receiving transition that activated the sending event $!A$ will fire.

Receiving messages B means deleting those messages from the corresponding channels. Therefore, those message channels of the state which activates that receiving event B are not empty. The successor state is calculated by subtracting B. Thus, the successor state set contains the successor states of those states only that activate the receiving event $?B$.

We will now define the complete interaction graph of an oWFN.

Definition 7 (Interaction Graph of an oWFN)
Let $N = (P, T, F)$ be an oWFN. The *interaction graph of* N is the directed graph $IG(N) = [R_N(\mathcal{A}(N)), B_N]$, with the root node $\mathcal{A}(N) = \{z \mid m_0 \xrightarrow{*} z\}$, the set of edges $B_N = \{[\mathcal{Z}, E, \mathcal{Z}'] \mid \mathcal{Z}, \mathcal{Z}' \in R_N(\mathcal{A}(N)) \wedge \mathcal{Z} \xrightarrow{E} \mathcal{Z}'$ with $E \in (\mathcal{R}(\mathcal{Z}) \cup \mathcal{S}(\mathcal{Z}))\}$ and $R_N(\mathcal{Z}) := \{\mathcal{Z}' \mid \mathcal{Z} \xrightarrow{*} \mathcal{Z}'\}$.

The node $V \in R_N(\mathcal{A}(N))$ is a *terminal node*, iff $\mathcal{R}(V) = \mathcal{S}(V) = \emptyset$. ⋆

For simplicity we will write $V \in IG(N)$ instead of $V \in R_N(\mathcal{A}(N))$, meaning V is a node in $IG(N)$.

3.1 Controllability in Interaction Graphs

The interaction graph of an oWFN depicts all possible states an oWFN can reach due to sending- and receiving events. We will now concentrate on answering the question about whether an oWFN is controllable. So, is it possible from the point of view of the controller that a sequence of sending- and receiving events will lead the net to terminate correctly? That means, is it possible starting from its initial state that the net will reach a final marking?

Control Strategy A *control strategy* is a sequence of sending- and receiving events such that the net reaches a final marking.

We will now classify the end nodes of an interaction graph formally.

Definition 8 (Good/ Bad End Node of an IG)
Let V be a terminal node by Def. 7 in the interaction graph of an oWFN and let Ω be the set of the final markings of the open workflow net. We call V a (i) *good end node*, iff $\forall z \in V : z \xrightarrow{*} \Omega' \in \Omega$ holds and (ii) *bad end node*, otherwise. $\star$

The interaction graph in Fig. 1(b) has three different end nodes. The end node $\{[c, p3], [p4]\}$ is a good end node. Its state set consists of two states – $[c, p3]$ is transient leading to state $[p4]$ and $[p4]$ is maximal and the final marking of the net. The end nodes $\{[b, c, p3], [b, p4]\}$ and $\{[c, p4]\}$ are bad end nodes. Both nodes do not activate any events. Further, there is no maximal state that is a final marking of the net.

Control Strategy in the Interaction Graph. As we look for a control strategy in the interaction graph we intuitively try to find a way from the root node to the good end nodes of the graph. The root node is just the initial state of the net and the good end nodes are basically those states indicating the net has weakly terminated. If there are bad end nodes we have to classify all nodes of the graph systematically. We hereby differentiate two types of nodes.

Definition 9 (Good Node)
Let V be a node in $IG(N)$. (i) If V is a good end node by Def. 8 in $IG(N)$, then V is a *good node*. (ii) If V is not a terminal node by Def. 7 in $IG(N)$ and there is at least one event activated in each maximal state of V that leads to a good node, then V is a *good node*. $\star$

The node $\{[c, p1], [c, p3, y], [p4, y]\}$ is a good node in the IG of Fig. 1(b). There is one maximal state in that node – $[p4, y]$. This state activates the receiving event ?y that leads to a good end node of the graph.

Definition 10 (Bad Node)
Let V be a node in $IG(N)$. V is a *bad node*, if at least one of the following properties is true for V: (i) V is a bad end node by Def. 8. (ii) There is at least one maximal state $z \in V$, such that all events activated in z lead to a bad node. (iii) All outgoing edges of V lead to a bad node. (iv) All incoming edges of V come from bad nodes. $\star$

Let's take a look at Fig. 1(b) again. There is a node $\{[p4]\}$. $[p4]$ is a final marking of the net. Hence, we could assume this node to be a good end node. However, during the analysis we declare this node to be a bad end node. This node has only one incoming edge ?z. This edge comes from node $\{[b, p1], [b, p3, y], [p2], [p4, z]\}$. The maximal state $[b, p3, y]$ activates two events – ?y and !c. Since both events lead to a bad end node, the node is declared to be a bad node. Therefore the node $\{[p4]\}$ is a bad node as well.

As we can easily see in the example above, the analysis of the nodes might identify a node to be good at first and declaring it to be bad later on. In the following we will also classify the states of a node.

Definition 11 (Good State)
Let V be a good node in the interaction graph $IG(N)$. Every maximal state z of V is a *good state*: (i) If V is a good end node by Def. 8, then $z = \Omega' \in \Omega$ holds. (ii) If V is not a terminal node by Def. 7, then $\exists e \in (\mathcal{R}(z) \cup \mathcal{S}(z)) : V \xrightarrow{e} V'$ and V' is a good node. $\star$

The maximal state $[b, p3, y]$ of the example above is not a good state of the node $\{[b, p1], [b, p3, y], [p2], [p4, z]\}$. However, the maximal state $[p3]$ of node $\{[p3]\}$ in Fig. 1(b) is a good state since it activates the sending event !c which leads to a good end node.

We will now define a subgraph in the interaction graph – the control strategy.

Definition 12 (Control Strategy in the Interaction Graph)
Let N be an oWFN and let $IG(N)$ be its interaction graph with the root node $\mathcal{A}(N)$. Let $IG_c(N) \sqsubseteq IG(N)$ be a subgraph of $IG(N)$ with the set of nodes V_c. The graph $IG_c(N)$ is called *control strategy of* N, if it holds:

(i) The node $\mathcal{A}(N)$ is the root node of $IG_c(N)$.
(ii) The leafs of the subgraph $IG_c(N)$ are good end nodes in $IG(N)$.
(iii) For every node $v \in V_c$ and every maximal state $z \in v$ there exists an event, which is activated in z and leads to a good successor node of v in $IG_c(N)$. $\star$

The control strategy[1] of an interaction graph consists of good nodes only. Starting from every node of the control strategy we can always reach a good end node. Therefore we can consider the control strategy as an operating guideline for the respective oWFN. With the help of the control strategy the controller can control the net in a way that it eventually terminates weakly.

In case of our example depicted in Fig. 1(b) we can conclude that the oWFN N1 is controllable. By analyzing the interaction graph we can find a control strategy. The controller has two options: sending a c and then receiving a y or first receiving y and then sending a c. Both ways lead the oWFN N1 to terminate weakly.

3.2 Complexity of Interaction Graphs

Interaction graphs were developed based on the reachability graphs of Petri nets. Those graphs as well as the interaction graphs suffer from the so-called *state-space-explosion* [Val88]. The number of states that a net can be in is extremely huge. As far as the interaction graphs are concerned we can see two dimensions of complexity – the number of nodes (comparable to the reachability graphs of Petri nets) and the size of the nodes. That is the size of the state sets.

Proposition 1 (Size of the Successor State Set): Let $\mathcal{Z}$ be a state set of an oWFN. Let $!A = \{a_1, \dots, a_n\}$ be a sending event and let $?B = \{b_1, \dots, b_m\}$ be a receiving event. The following holds for the succssor state set $\mathcal{Z}'$.

(i) *sending event*: $\exists \mathcal{Z}' : \mathcal{Z} \xrightarrow{!a_1,\dots,a_n} \mathcal{Z}' \Rightarrow \mathcal{Z} + a_1 + \dots + a_n \subset \mathcal{Z}'$
(ii) *receiving event*: $\exists \mathcal{Z}' : \mathcal{Z} \xrightarrow{?b_1,\dots,b_m} \mathcal{Z}' \Rightarrow \mathcal{Z} \supseteq \mathcal{Z}' + b_1 + \dots + b_m$ $\star$

The notation $\mathcal{Z} + x$ represents the set $\{z + x \mid z \in \mathcal{Z}\}$.

Proof (Proposition 1): (i) It is to show $\mathcal{Z} + a_1 + \dots + a_n \subset \mathcal{Z}'$. Let $!A$ be the activated sending event for which the successor node is being calculated. By Def. 6 it holds $\mathcal{Z} + a_1 + \dots + a_n \subseteq \mathcal{Z}'$. By Def. 4 there exists at least one $z \in \mathcal{Z}$ with $z + A \xrightarrow{?A} z'$ and $z[A] = 0$, that activates the sending event $!A$. By Def. 6 we can find a state z' in the state set $\mathcal{Z}'$ of the successor node for which $z + A \xrightarrow{?A} z'$ holds. It holds for state z' $z'[A] = 0$, because transition $?A$ has fired. Therefore we have $z' \in \mathcal{Z}'$ and $z' \notin \mathcal{Z} + A$.

[1] In this paper the control strategy of an interaction graph is depicted as a solid line. Those nodes and edges that do not belong to the control strategy of the graph are drawn with dashed lines.

(ii) Let $!B$ be a receiving event. It is to show: $\mathcal{Z}' + B \subseteq \mathcal{Z}$. Let $z \in \mathcal{Z}' + B$ be arbitrarily chosen. Then $z - B \in \mathcal{Z}'$ holds. By Def. 6 we have $\mathcal{Z}' = \{z - B \mid z \in \mathcal{Z} \wedge z[B] \geq 1\}$. Consequently we have $z \in \mathcal{Z}$. □

Sending events let the size of the successor node grow by at least one more state. Just the receiving events can reduce the size of the successor node.

4 Reduction of Interaction Graphs

In this section we will introduce reduction rules that will reduce the number of events being considered for building up the interaction graph. For every reduced interaction graph we will proof that it still can be used for showing that the associated net is controllable. At the end we will define a (reduced) interaction graph that combines all reduction rules in a certain way and will then be the basis for an efficient computation of a control strategy and therefore can be used for the analysis of controllability of a net.

By constructing the whole interaction graph we have considered all activated sending events and receiving events. For showing that a net is controllable we do not necessarily have to consider all possible events. In the following we will show rules that tell us which activated events are necessary for the computation of the interaction graph.

We will use the following naming convention: Every interaction graph will be abbreviated by $IG_X(N)$ with X being a shortcut for the rule. Further, we define new sets of activated events that we will name *reduced activated sending (receiving) events*. Hereby, we again use the X as a subscript to indicate the associated rule. The set of all maximal states of a node in the interaction graph will be called $\mathcal{Z}_{max}(V)$ with $\mathcal{Z}_{max}(V) = \{z \mid z \in V \wedge \text{ z is maximal}\}$.

4.1 Transient States

It is possible that there are transient states within a node of the interaction graph. Being in a transient state the net can change to another state without letting its controller know. So, just the maximal states supply a surety in a certain way. Suppose a transient state activates a sending event. If the controller now sends a message to the net, there is no way of knowing that the net will ever consume this message. The net might just have switched to another state, which does not activate this event anymore. So, we will only consider the maximal states of a node to compute the activated events.

Example 2. Let us take a look at Fig. 1 again. It shows the oWFN N1 (Fig. 1(a)) and its complete interaction graph IG(N1) (Fig. 1(b)). There are two different states in the root node of the interaction graph – [p1] (transient) and [p3, y] (maximal). During the calculation of the complete interaction graph (Fig. 1(b)) the sending event !b was considered. This sending event is only activated in the transient state [p1]. The maximal state [p3, y] activates the sending event !c and the receiving event ?y. For the calculation of the reduced interaction graph (see Fig. 2) we now use the sending event !c and the receiving event ?y only. ⋆

Definition 13 (Reduction by Maximal States)
Let N be an oWFN with $\mathcal{A}(N)$ being the set of its initial states. The reduced interaction graph $IG_{max}(N)^2$ of N is a directed graph $[V, E]$ with nodes V and edges E. It is defined inductively as follows.

[2] *max* stands for "**Max**imal States".

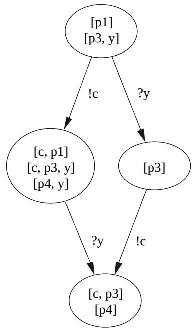


Fig. 2. Reduced IG of N1 where only maximal states are considered.

(i) The set of states $\mathcal{A}(N)$ is the root node of $IG_{max}(N)$.
(ii) If $K \in IG_{max}(N)$, K is not a terminal node by Def. 7 and there is an $M \in (\mathcal{R}(\mathcal{Z}_{max}(K)) \cup \mathcal{S}(\mathcal{Z}_{max}(K)))$, then K' with $K \xrightarrow{M} K'$ is a node and $[K, M, K']$ is an edge in $IG_{max}(N)$. $\star$

Proposition 2: Let N be an oWFN. If $IG(N)$ contains a control strategy, then there is a control strategy in $IG_{max}(N)$. $\star$

Proof (Proposition 2): Let $IG(N)$ be an interaction graph of the oWFN N with the control strategy $IG_c(N)$ by Def. 12 and let $IG_{max}(N)$ be the reduced interaction graph of N by Def. 13. It is to show that there exists a control strategy $IG_{max_c}(N)$ in $IG_{max}(N)$.

The root node of the graphs $IG(N)$, $IG_c(N)$ and $IG_{max}(N)$ are equal by Def. 13. We set the root node of $IG_{max_c}(N)$ equal to the root node of $IG_c(N)$. For every node V with $V \in IG_c(N)$ and $V \in IG_{max}(N)$ and V is not a terminal node by Def. 7, those successor nodes of V from $IG_c(N)$ will be taken into $IG_{max_c}(N)$, that are computed by events being activated in maximal states of V.

We claim that $IG_{max_c}(N)$ is a control strategy in $IG_{max}(N)$. The root node of $IG_{max_c}(N)$ is equal to the root node of $IG_{max}(N)$ and the root node of $IG_c(N)$ by construction. By Def. 12 it holds: if $V \in IG_c(N)$ and V is not a terminal node, then for every maximal state in V there exists at least one event, which leads to a node in $IG_c(N)$. Let $\mathcal{V}' \in IG_c(N)$ be the set of successor nodes of V in $IG_c(N)$, which are computed by events, that are activated in the maximal states of V. Node V is in $IG_{max}(N)$ by construction. The successor nodes $\mathcal{V}'$ are in $IG_{max}(N)$ as well, because they were computed by events, that were activated in maximal states. By construction the nodes of $\mathcal{V}'$ were put into $IG_{max_c}(N)$. Therefore it holds: If $K \in IG_{max_c}(N)$ and K is not a terminal node, there exists for every maximal state in K at least one event, which leads to a node in $IG_{max_c}(N)$. Because just nodes from $IG_c(N)$ were taken into $IG_{max_c}(N)$ and the end nodes of $IG_{max}(N)$ are terminal nodes, the end nodes of $IG_{max_c}(N)$ are good end nodes.

Therefore we can conclude, that there exists a control strategy $IG_{max_c}(N)$ in $IG_{max}(N)$, if there is a control strategy $IG_c(N)$ for $IG(N)$. By construction $IG_{max_c}(N) \sqsubseteq IG_c(N)$ holds. $\square$

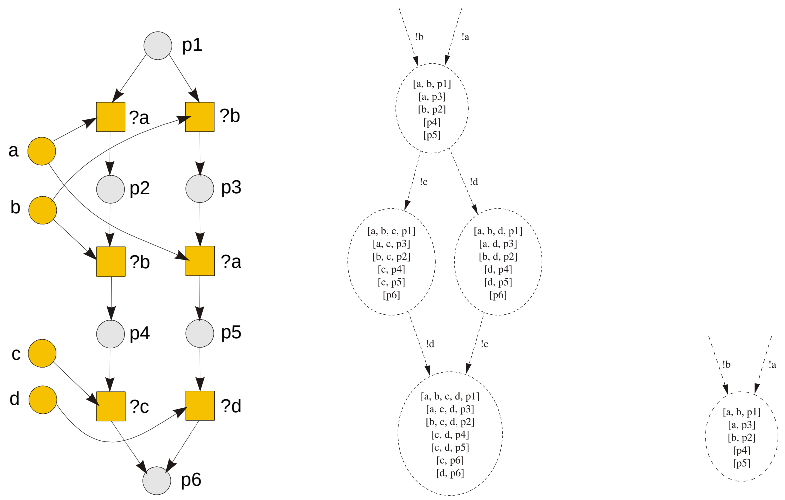
We have shown, that we can use the reduced interaction graph by rule 13 for the controllability analysis. Therefore we will integrate this rule into the following reduction rules. Thus, we will only consider the maximal states of a node to compute the activated events for the computation of the successor nodes.

4.2 Left Behind Messages

One of the main reasons for a node to be assigned bad during the computation of the control strategy (see Sec. 3.1), is that messages remain in the message channels. The net eventually terminates with some of its channels not being empty. In this section we will take a closer look at each node and its maximal states in order to figure out if there is a maximal state which leads to a bad end node. Therefore we first determine which sending events are activated. If all maximal states activate the same sending events, the associated messages will be consumed right away. So, the more interesting case is the one where there are messages being sent that cannot or will not be consumed immediately. With the analysis of the structure of the node we want to achieve two things:

(i) Early identification of those nodes, that will be classified as bad nodes during the computation of the control strategy later on (refer to example 3(a)).
(ii) Setting up a black list of sending events, that will eventually lead to bad nodes (refer to example 3(b)).

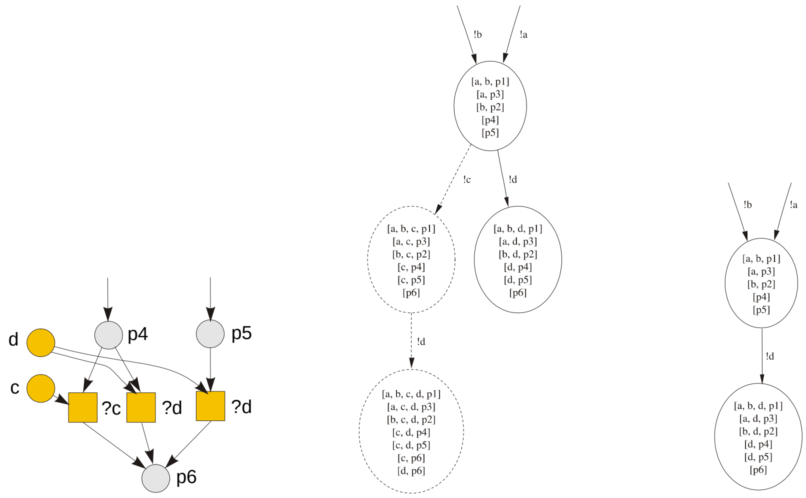
Example 3. (a) Figure 3(a) depicts the oWFN `N2`. A part of the corresponding interaction graph is shown in Fig. 3(b). Let's take a look at the state set of the first node of the graph. The net can either be in state `p4` or in state `p5`. Both maximal states activate different sending events. As we go on in the graph we can easily see that each message which cannot be consumed by the other state right away will stay in the message channel. There is no upcoming receiving transition that can consume the respective message.
The idea behind the reduction rule in this section is not to consider the sending events `!c` and `!d` for the calculation of successor nodes. Therefore we get the reduced interaction graph of `N2` that is depicted in Fig. 3(c).



(a) oWFN N2 (b) Part of the IG of N2 (c) Part of the reduced IG of N2

Fig. 3. Left behind messages – I.

(b) We now extend the oWFN `N2` in a way that the state `p4` activates the sending event `!d` as well (see Fig. 4(a)). The state `p5`, however, still activates the sending event `!d` only. As we can tell from the interaction graph (see Fig. 4(b)), the sending of message `d` now leads to a good end node. The message can be consumed by both maximal states `p4` and `p5`. But the message `c` stays in the message channel and therefore the sending event `!c` leads to a bad end node. Because of the analysis of the node, we do not calculate the successor nodes of the sending event `!c` anymore. The corresponding reduced interaction graph is shown in Fig. 4(c). ⋆



(a) Part of oWFN N3 (b) Part of the IG of N3 (c) Part of the reduced IG of N3

Fig. 4. Left behind messages – II.

During the early identification of nodes and the creation of the black list we will restrict ourselves to considering only those states of a node, that are maximal and that activate sending events only. We will call this set of states Z_{ms}.

Early Detection. For every state of Z_{ms} we compute the set of activated sending events. Now, for every sending event we check if the message sent can be consumed by the other maximal states of the node. Note, that messages do not necessarily have to be consumed right away. They can be consumed later on as well and thus will remain in the message channels at first. If we are able to find a maximal state in Z_{ms}, that activates sending events which will never be consumed by another maximal state of the node, we will classify this node as bad. The definition of the control strategy requires for every maximal state at least one event that leads to a good node. But we have found one maximal state, that activates just sending events, which will never be consumed by all other maximal states. Therefore no good end node is possible anymore.

In example 3(a) the first node of the IG in Fig. 3(b) was detected to be bad. Therefore its successor nodes were not calculated anymore (see Fig. 3(c)).

Black List. During the early detection of bad nodes we successively check all sending events for every state in Z_{ms}. It surely is possible that a state in Z_{ms} activates more than one sending event. The control strategy requires that at least one sending event will lead to a good node. That is why, we want to create a black list of sending events, that will not lead to a good node. So, should we find a sending event during the early detection analysis that will not be eventually consumed by all maximal states of the node, we will add this event to the black list of the node. As we compute the successor nodes of the current node, we will not consider the events of the black list anymore.

Let's take a look at example 3(b). The black list of the first node of the IG shown in Fig. 4(b) is $\{c\}$. Thus we do not consider the sending event $!c$ for calculating successor nodes (see Fig. 4(c)).

As we do the early detection analysis and the computation of the black list we have to find out whether certain messages can be consumed by a state. That means, if a state is not able to consume a message right away, we have to find out, if this message can be consumed later on. We will define the set of messages that a state can consume in the future. Before, we have to define the set of all successor transitions of a state.

Definition 14 (Set of Successor Transitions)
Let $p \in P$ be a place in the Petri net $N = (P, T, F)$. The *set of successor transitions* $\mathcal{T}(p)$ of place p contains all transitions being reachable from p and is defined inductively:

A) $\mathcal{T}(p) = \{t \mid t \in p^\bullet\}$
I) If $t \in \mathcal{T}(p)$ and $t' \in (t^\bullet)^\bullet$, then $t' \in \mathcal{T}(p)$. ⋆

Definition 15 (Consumable Messages of a State)
Let N be an oWFN with the interaction graph $IG(N)$. Let V be a node in $IG(N)$. The set of the *consumable messages* $CM(z)$ of state $z \in V$ is defined as follows.

$$CM(z) = \{\mu(t) \mid p \in z \cap inner(N) \wedge t \in \mathcal{T}(p)\}. \quad \star$$

The function $\mu(t) = ({}^\bullet t \cap in(N)) \cup (t^\bullet \cap out(N))$ returns the name of the messages that the corresponding transition can produce or consume. With the help of the consumable messages of a state we abstract from the set of all successor transitions. We take a look at all internal places of a state z and check, which transitions are reachable from that place. We then add the corresponding message name of the transition to the set $CM(z)$.

Definition 16 (Black List)
Let N be an oWFN and let V be a node in the interaction graph of N. The set $BL(V)$ is the *black list* of node V.

$$\begin{aligned} BL(V) = \{i \mid\, & i \in \mathcal{S}(\mathcal{Z}_{max}(V)) && (1)\\ & \wedge \exists z_1, z_2 \in \mathcal{Z}_{max}(V) : i \in \mathcal{S}(z_1) \wedge i \notin \mathcal{S}(z_2) && (2)\\ & \wedge i \notin CM(z_2)\} && (3) \end{aligned}$$

⋆

The black list of sending events of a node V contains messages of sending events. These sending events are activated in V (1). There exists at least one state, which does not activate this event (2). This state will never consume the message associated with that event (3). Therefore, the message will remain in a message channel. That is why, we will add this event to the black list of the node.

Corollary 3: Let V be a node in the interaction graph $IG(N)$. Let $BL(V)$ be the black list of node V. Let $S \in BL(V)$. If $V \xrightarrow{S} V'$, then V' is not a good node. ⋆

Proof (Corollary 3): Let $S \in BL(V)$. By Def. 16 there exists at least one state in V that activates the sending event $!S$. Furthermore, there is at least one state $z_2 \in \mathcal{Z}_{max}(V)$ with $S \notin \mathcal{S}(z_2)$. By Def. 16 $S \notin CM(z_2)$ holds and by Def. 6 $z_2 + S \in V'$ holds.

By Def. 15 the set $CM(z_2)$ contains all sending events, that are associated with the receiving transitions reachable by z_2. Because $S \notin CM(z_2)$ holds, there exists no receiving transition for S, which is reachable from z_2. Therefore the message S cannot be consumed and will remain in the message channel for all successor states of z_2. So there is no sequence of events, which will lead $z_2 + S$ to a good end node. The successor states of z_2 are no good states if the sending event $!S$ occurs. Consequently, the maximal state $z_2 + S$ is no good state and the successor node V' with $V \xrightarrow{S} V'$ is no good node as well. $\square$

We have shown that the sending events of the black list of a node will not lead to good nodes. Hence, we will not compute the successor nodes of those sending events.

Definition 17 (Analysis of Sending Events)
Let N be an oWFN and let V be a node in the interaction graph $IG(N)$. The set of the reduced activated incoming messages $\mathcal{S}_{nsa}(V)^3$ is defined as follows.

$$\mathcal{S}_{nsa}(V) = \mathcal{S}(\mathcal{Z}_{max}(V)) \setminus BL(V). \qquad \star$$

The black list of sending events will also be a basis for the early detection analysis described above.

Definition 18 (Bad End Node by Node Analysis)
Let N be an oWFN and let V be a node in $IG(N)$. V is a *bad end node by node analysis*, iff

$$\exists z \in \mathcal{Z}_{max}(V) : \mathcal{R}(z) = \emptyset \wedge (\forall i \in \mathcal{S}(z) : i \in BL(V)). \qquad \star$$

We want to classify a node as a bad node, if we find a state in that node, that (1) activates no receiving events and (2) that activates solely sending events, which cannot be consumed by any other state of the node. That means, all activated sending events of that particular state are in the black list of the node.

Now we can define a reduced interaction graph which uses the results obtained by the node analysis and the sending event analysis.

Definition 19 (Reduction by Node Analysis and Sending Event Analysis)
Let N be an oWFN with $\mathcal{A}(N)$ being its set of initial states. The set of the reduced activated incoming messages $\mathcal{S}_{nsa}(K)$ of node K is computed according to Def. 17. The reduced interaction graph $IG_{nsa}(N)$ of N is a directed graph $[V, E]$ with nodes V and edges E and is defined inductively as follows.

(i) The set of states $\mathcal{A}(N)$ is the root node of $IG_{nsa}(N)$.
(ii) If $K \in IG_{nsa}(N)$ and
  - K is not a bad node by Def. 18 and K is not a terminal node by Def. 7 and there is a $M \in (\mathcal{R}(\mathcal{Z}_{max}(K)) \cup \mathcal{S}_{nsa}(K))$, then K' with $K \xrightarrow{M} K'$ is a node and $[K, M, K']$ is an edge in $IG_{nsa}(N)$.
  - K is a bad end node by Def. 18 no successor nodes will be computed. $\star$

The reduced interaction graph $\mathcal{S}_{nsa}(K)$ by Def. 19 possesses the same root node as the complete interaction graph. For the calculation of the successor nodes all receiving events being activated by maximal states are used with no restriction. The reduced activated sending events for each node are computed by Def. 17. If a node is classified as being a bad node by Def. 18 no successor nodes will be computed from that node.

Proposition 4: Let N be an oWFN. If there is a control strategy in $IG(N)$, then there is a control strategy in $IG_{nsa}(N)$. $\star$

[3] *nsa* stands for "**N**ode Analysis and **S**ending Event **A**nalysis"

PROOF (PROPOSITION 4): Let N be an oWFN with the interaction graph $IG(N)$. There exists a control strategy $IG_c(N)$ and the reduced interaction graph $IG_{nsa}(N)$ is computed. It is to show, that there exists a control strategy IG_{nsa_c} in $IG_{nsa}(N)$.

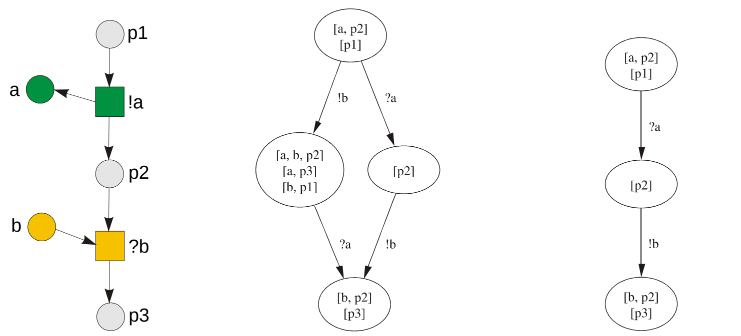
We claim that IG_{nsa_c} is equal to $IG_c(N)$. Let V be a node in $IG_c(N)$ and in $IG_{nsa}(N)$ with the black list $BL(V)$. By Corollary 3 all sending events in $BL(V)$ will lead to no good nodes. The events, that will lead from V to nodes in $IG_c(N)$ are therefore not contained in $BL(V)$. Hence, all successor nodes $\mathcal{V} \in IG_c(N)$ of V, that are computed by events being activated in the maximal states of V, are nodes of $IG_{nsa}(N)$ as well. From this it follows that $IG_{nsa_c} = IG_c(N)$ □

We have just shown that we can use the reduced interaction graph $IG_{nsa}(N)$ for the analysis of the controllability of an oWFN N.

4.3 Both kind of Events activated in one State

We turn our attention to those maximal states, that activate receiving events. If such a state activates a sending event as well, we let the receiving event occur first. The sending event will still be activated afterwards. Note, sending events with the same name can be activated by different receiving transitions. We therefore do not consider those sending events, that are activated by states that activate receiving events as well.

Example 4. In Fig. 5 we can see the oWFN N4 with its complete interaction graph (Fig. 5(b)) and its reduced interaction graph (Fig. 5(c)). The root node of the two



(a) oWFN N4

(b) Complete IG of N4

(c) Reduced IG of N4

Fig. 5. Receiving before sending.

graphs contains the maximal state [a, p2]. This state activates the receiving event ?a as well as the sending event !b. The controller now shall receive the message a before sending b. The sending event !b is still activated in the successor node. ⋆

In order to compute the reduced interaction graph we will now define the set of reduced activated sending events of a node of the graph, that fullfills the requirements we have just described.

Definition 20 (Receiving before Sending)
Let N be an oWFN and let $V \in IG(N)$. The set of the reduced activated incoming messages $\mathcal{S}_{rbs}(V)$[4] is defined as follows.

$$\mathcal{S}_{rbs}(V) = \{i \mid z \in \mathcal{Z}_{max}(V) \wedge i \in \mathcal{S}(z) \wedge \mathcal{R}(z) = \emptyset\}. \quad \star$$

[4] *rbs* stands for "**R**eceiving **B**efore **S**ending".

For every maximal state of V we check if it activates a sending event and no receiving event. If this is the case, we will add the activated sending event to the set $\mathcal{S}_{rbs}(V)$. Consequently, just those sending events are considered to be active that are activated by states that do not activate any receiving events.

Now we can define the reduced interaction graph based on this rule.

Definition 21 (Reduction by Receiving before Sending)
Let N be an oWFN with $\mathcal{A}(N)$ being its set of initial states. The set of the reduced activated incoming messages $\mathcal{S}_{rbs}(K)$ of node K is computed by Def. 20. The reduced interaction graph $IG_{rbs}(N)$ of N is a directed graph $[V, E]$ with nodes V and edges E and is inductively defined as follows.

(i) The set of states $\mathcal{A}(N)$ is the root node of $IG_{rbs}(N)$.
(ii) If $K \in IG_{rbs}(N)$, K is no terminal node by Def. 7 and there is a $M \in (\mathcal{R}(\mathcal{Z}_{max}(K)) \cup \mathcal{S}_{rbs}(K))$, then K' with $K \xrightarrow{M} K'$ is a node and $[K, M, K']$ is an edge in $IG_{rbs}(N)$. $\star$

We will now show, that the reduced interaction graph by Def. 21 can be used for the controllability analysis of an oWFN.

Proposition 5: Let N be an oWFN. If there is a control strategy in $IG(N)$, then there is a control strategy in $IG_{rbs}(N)$. $\star$

For proofing Proposition 5 we need a few more corollaries and lemmas, that we will state and proof in the following.

Corollary 6: Let N be an oWFN with the interaction graph $IG(N)$ and let K be a node in $IG(N)$. If $K \xrightarrow{S} K'$ and $S \in \mathcal{S}(K)$, then $\mathcal{R}(K) \subseteq \mathcal{R}(K')$ is valid. $\star$

Proof (Corollary 6): Consider $K \xrightarrow{S} K'$ with $S \in \mathcal{S}(K)$. Let E be an arbitrary receiving event out of the set $\mathcal{R}(K)$. To show: $E \in \mathcal{R}(K')$. E is a receiving event. Hence, by Def. 4 $\exists z \in K : z[E] \geq 1 \wedge E \subseteq out(N)$ holds.

1. $S \notin \mathcal{S}(z)$. By Def. 6 $z + S \in K'$ is true. Hence, $\mathcal{R}(K) \subseteq \mathcal{R}(K')$.
2. $S \in \mathcal{S}(z)$. By Def.1 $\neg\exists t \in T : p \in {}^\bullet t \wedge p \in out(N)$ holds. Furthermore, by Def. 6: $\mathcal{Z}' = \{z' \mid z + S \xrightarrow{*} z'\} \subseteq K'$. By computing the set $\mathcal{Z}'$ no tokens are taken away from places in $out(N)$. So, it holds $\exists z' \in K' : z'[E] \geq 1$ and therefore: $\mathcal{R}(K) \subseteq \mathcal{R}(K')$. $\square$

Corollary 7: Let N be an oWFN with the interaction graph $IG(N)$ and let K be a node in $IG(N)$. Let z be a state in K. If $z \xrightarrow{E} z'$ and $E \in \mathcal{R}(z)$, then $\mathcal{S}(z) = \mathcal{S}(z')$. $\star$

Proof (Corollary 7): Let $z \xrightarrow{E} z'$ and $E \in \mathcal{R}(z)$. We need to show: $\mathcal{S}(z) = \mathcal{S}(z')$.

1. $\mathcal{S}(z) \subseteq \mathcal{S}(z')$. Let $S \in \mathcal{S}(z)$ be arbitrarily selected. We show: $S \in \mathcal{S}(z')$. By assumption and by Def. 6 $z' = z - E$, $z[E] \geq 1$ and $E \subseteq out(N)$ hold. Furthermore we have $z[p] = z'[p]$ for all $p \in in(N) \cup inner(N)$. Consequently, $S \in \mathcal{S}(z')$.
2. $\mathcal{S}(z) \supseteq \mathcal{S}(z')$. Let $S \in \mathcal{S}(z')$ be arbitrarily selected. We show: $S \in \mathcal{S}(z)$. We have by Def. 6: $z' + E = z$. The sending event S is activated in z'. By Def. 4 $\exists z'' : z' + S \xrightarrow{?S} z'' \wedge z'[S] = 0$. Because of $E \subseteq out(N)$, $z[S] = 0$ is valid for z and $z + S$ activates the receiving transition $?S$. Therefore $S \in \mathcal{S}(z)$ holds. $\square$

Corollary 8: Let K be a node in the interaction graph $IG(N)$. Let $z \in K$ be a state and let $E \in \mathcal{R}(z)$. If $z \xrightarrow{E} z'$, then $\mathcal{R}(z - E) = \mathcal{R}(z')$. $\star$

Proof (Corollary 8): We will show: $\mathcal{R}(z-E) = \mathcal{R}(z')$. Because $z \xrightarrow{E} z'$ with $E \in \mathcal{R}(z)$ holds, we have $z' = z - E$ by Def. 6. Consequently we have $\mathcal{R}(z') = \mathcal{R}(z-E)$. $\square$

Lemma 9: Let K be a node in the interaction graph $IG(N)$. Let $z \in K$ be a state and let $E \in \mathcal{R}(z)$ and $S \in \mathcal{S}(z)$. If $K \xrightarrow{E} K' \xrightarrow{S} K''$ and $K \xrightarrow{S} K^* \xrightarrow{E} K^{**}$, then $K'' \subseteq K^{**}$. $\star$

Proof (Lemma 9): It is to show: $K'' \subseteq K^{**}$. Let $z'' \in K''$ be arbitrarily selected. We will show $z'' \in K^{**}$. We have by Def. 6 $K'' = \{z'' \mid z' \in K' \wedge z' + S \xrightarrow{*} z''\}$. Hence, there exists a $z' \in K'$ with $z' + S \xrightarrow{*} z''$. By Proposition 1 $z' + E = z \in K$ holds. If $K \xrightarrow{S} K^*$, then $z + S \xrightarrow{*} z^*$ and $z^* \in K^*$ and in particular $z' + E + S \xrightarrow{*} z^*$ and $z^* \in K^*$ hold. Because $E \subseteq out(N)$ and E does not activate or deactivate a transition, we can find a $z^* \in K^*$, such that $z^* = z'' + E$. We now choose a $z^* \in K^*$, which has this property. If $K^* \xrightarrow{E} K^{**}$ holds, we have by Def. 6 $z^* - E = z'' \in K^{**}$. $\square$

Lemma 10: Let K be a node in the interaction graph $IG(N)$. Let $z \in K$ be a state and let $E_1, E_2 \in \mathcal{R}(z)$. If $K \xrightarrow{E_1} K' \xrightarrow{E_2} K''$ and $K \xrightarrow{E_2} K^* \xrightarrow{E_1} K^{**}$, then $K'' = K^{**}$ holds. $\star$

Proof (Lemma 10): It is to show: $K'' = K^{**}$.

1. To show: $K'' \subseteq K^{**}$. Let $z'' \in K''$ be arbitrarily selected. We will show: $z'' \in K^{**}$. By Proposition 1 $z'' + E_2 \in K'$ and $z'' + E_2 + E_1 \in K$ hold. By Def. 6 $(z'' + E_2 + E_1) - E_2 \in K^*$ and $(z'' + E_1) - E_1 \in K^{**}$ hold as well.
2. To show: $K^{**} \subseteq K''$. Let $z^{**} \in K^{**}$ be arbitrarily selected. We will show: $z^{**} \in K''$. By Proposition 1 $z^{**} + E_1 \in K^*$ and $z^{**} + E_1 + E_2 \in K$ hold. By Def. 6 $(z^{**} + E_1 + E_2) - E_1 \in K'$ and $(z^{**} + E_2) - E_2 \in K''$ hold as well. $\square$

Lemma 11: Let K and T be two nodes in the interaction graph $IG(N)$ with $T \subseteq K$. If K is a good node in $IG(N)$, then T is a good node in $IG(N)$ as well. $\star$

Proof (Lemma 11): K is a good node. Therefore the properties defined in Def. 9 hold for K. Let $[\omega] \subseteq \Omega$ be a final marking of the oWFN N.

(i) Let K be a good end node. By Def. 8 $\forall z \in K : z \xrightarrow{*} [\omega]$ holds for K. There exists exactly one maximal state in K: $[\omega]$. Since $T \subseteq K$ and T is a node in $IG(N)$, there is exactly this maximal state $[\omega]$ in T as well. By Def. 3 all transient states of T lead to $[\omega]$. Therefore, T is a good end node.

(ii) Let K not be a terminal node in $IG(N)$. Since K is a good node, there exists at least one event for every maximal state in K, which leads to a good node. Let $z \in \mathcal{Z}_{max}(T)$ be chosen arbitrarily. Because of $T \subseteq K$ there exists one $X \in (\mathcal{R}(z) \cup \mathcal{S}(z))$, such that $K \xrightarrow{X} K'$. We distinguish between two cases for X:
   (a) $X \in \mathcal{R}(z)$. By Def. 6: $K' = \{z - X \mid z \in K \wedge z[X] \geq 1\}$. Since $z \in T$, $T \subseteq K$ and T is a node by Def. 3, it holds that $T' = \{z - X \mid z \in T \wedge z[X] \geq 1\} \subseteq K'$ and T' is a node by Def. 3.
   (b) $X \in \mathcal{S}(z)$. By Def. 6: $K' = \{z' \mid z \in K : z + X \xrightarrow{*} z'\}$. Since $z \in T$, $T \subseteq K$ and T is a node by Def. 3, it holds that $T' = \{z' \mid z \in T : z + X \xrightarrow{*} z'\} \subseteq K'$ and T' is a node by Def. 3. $\square$

The following Lemma 12 plays a central role in the proof of Proposition 5. We will show that every receiving event being activated in a good node leads from that good node to a good successor node.

Lemma 12: Let N be an oWFN with the interaction graph $IG(N)$. Let K be a node in the control strategy $IG_c(N)$. All receiving events $\mathcal{R}(K)$ lead to a node in $IG_c(K)$. $\star$

Proof (Lemma 12): Assume there exists a good node $K \in IG_c(N)$ with $K \xrightarrow{E} K'$, K' is a bad node and $E \in \mathcal{R}(K)$. We choose $K \in IG_c(N)$ in such a way, that $K \xrightarrow{E} K'$ is the last such sequence of nodes in $IG_c(N)$. By last sequence we mean, that there are no nodes K_1 and K_1', such that $K_1 \xrightarrow{E} K_1'$ with $K_1 \in IG_c(N)$ is a good node and K_1' is a bad node and $K \xrightarrow{*} K_1$. Since both $IG(N)$ and $IG_c(N)$ are acyclic and finite, there exists such a last sequence.

Because K' is a bad node, K' fullfills at least one of the properties of Def. 10. We will show, that those properties of K' lead to a contradiction.

(i) Let K' be a bad end node. Hence, $\mathcal{S}(K') = \mathcal{R}(K') = \emptyset$ is true by Def. 7. Let $z' \in K'$ be arbitrarily selected. It holds: $\mathcal{R}(z') = \mathcal{S}(z') = \emptyset$. By Proposition 1 $z' + E = z \in K$ holds. By Corollary 7 $\mathcal{S}(z') = \mathcal{S}(z) = \emptyset$ holds and by Corollary 8 $\mathcal{R}(z - E) = \mathcal{R}(z') = \emptyset$ is valid. There exists no event for state z, which leads to a good node. Therefore K is no good node. This is a contradiction to our assumption.

(ii) There exists at least one state in K', such that all events being activated in that state lead to a bad node. Let $z' \in K'$ with $K' \xrightarrow{X} K''$ and K'' is a bad node and $X \in (\mathcal{R}(z') \cup \mathcal{S}(z'))$ be chosen arbitrarily. By Proposition 1 $z' + E = z \in K$ holds. Since K is a good node and since the receiving event $?E$ leads to a bad node from K, there exists an event $X \in (\mathcal{R}(z) \setminus \{E\} \cup \mathcal{S}(z))$ with $K \xrightarrow{X} K^*$ and K^* is a good node. We distinguish between two cases for X:

   (a) $X \in \mathcal{R}(z)$. By Corollary 8 the receiving event $E \in \mathcal{R}(z)$ is still activated in K^*. By Lemma 10 $K \xrightarrow{E} K' \xrightarrow{X} K''$ and $K \xrightarrow{X} K^* \xrightarrow{E} K''$ hold. K'' is a bad node after our premise. This is, however, a contradiction to our assumption, that $K \xrightarrow{E} K'$ is the last such sequence of nodes in $IG(N)$.

   (b) $X \in \mathcal{S}(z)$. By Corollary 6 the receiving event $E \in \mathcal{R}(z)$ in K^* is still activated. By Lemma 9 the following holds: if $K \xrightarrow{E} K' \xrightarrow{X} K''$ and $K \xrightarrow{X} K^* \xrightarrow{E} K^{**}$, then K'' is a subset of K^{**}. We assumed that K'' is a bad node. Hence, K^{**} is a bad node by Lemma 11. There exists another sequence of type $K \xrightarrow{E} K'$. It actually is $K^* \xrightarrow{E} K^{**}$. This is a contradiction to our assumption.

(iii) All outgoing edges from K' lead to bad nodes. That means, all events being activated in K' lead to bad nodes. Let the state $z' \in K'$ be chosen arbitrarily. Therefore it holds for all $X \in (\mathcal{R}(z') \cup \mathcal{S}(z'))$, that if $K' \xrightarrow{X} K''$, then K'' is a bad node. We will now use the same argumentation for z' as we have done for z' in (ii).

(iv) Contradiction since K is a good node after assumption.

We have assumed that $K \xrightarrow{E} K'$ with K being a good node and with K' being a bad node and $E \in \mathcal{R}(K)$ is the last such sequence of nodes in $IG_c(N)$. We have shown by items (ii).a, (ii).b and (iii).a, (iii).b that the sequence of nodes chosen cannot be the last sequence with the properties assumed in $IG_c(N)$. By item (i) we have shown, that if K' is a bad node, that K necessarily has to be a bad node as well. So, we can conclude that there exists no sequence of nodes of type $K \xrightarrow{E} K'$ with K being a good node and K' is a bad node and $E \in \mathcal{R}(K)$. That is why, all receiving events $E \in \mathcal{R}(K)$ have to lead to good nodes. $\square$

Now we are ready to proof Proposition 5.

PROOF (PROPOSITION 5): There exists a control strategy $IG_c(N)$ in the interaction graph $IG(N)$ of the oWFN N. Let $IG_{rbs}(N)$ be the reduced interaction graph of N by Def. 21. It is to show that there exists a control strategy $IG_{es_c}(N)$ in $IG_{rbs}(N)$.

The root nodes of the graphs $IG(N)$, $IG_c(N)$ and $IG_{rbs}(N)$ are equal by Def. 21. Let the root node of $IG_{es_c}(N)$ be equal to the root node of $IG_c(N)$. For every node V with $V \in IG_c(N)$ and $V \in IG_{rbs}(N)$ and V is not a terminal node by Def. 7, we add the successor nodes of V from $IG_c(N)$ to $IG_{es_c}(N)$, that are computed by events $E \in \mathcal{R}(\mathcal{Z}_{max}(V)) \cup \mathcal{S}_{rbs}(V)$.

We claim $IG_{es_c}(N)$ is a control strategy in $IG_{rbs}(N)$. The root nodes of $IG_{es_c}(N)$, $IG_{rbs}(N)$ and $IG_c(N)$ are equal by construction. If V is a node in $IG_c(N)$ and if $\mathcal{S}_{rbs}(V) = \mathcal{S}(V)$ holds, then all successor nodes of V from $IG_c(N)$ were taken into $IG_{es_c}(N)$. If on the other hand $\mathcal{S}_{rbs}(V) \subset \mathcal{S}(V)$ holds, those successor node of V in $IG_c(N)$, that can be reached by events $\mathcal{S}(V) \setminus \mathcal{S}_{rbs}(V)$, were not added to $IG_{es_c}(N)$. By Def. 20 those events $\mathcal{S}(V) \setminus \mathcal{S}_{rbs}(V)$ are only activated by states in V, that activate receiving events as well. By Lemma 12 all receiving events of V lead to nodes in $IG_c(N)$. If V is not a terminal node by Def. 7, there exists for every maximal state in V at least one event, that leads to a node in $IG_c(N)$ and therefore to a node in $IG_{es_c}(N)$ as well. If V is a terminal node by Def. 7, then V is a good end node (Def. 8) by construction.

Consequently, there exists a control strategy $IG_{es_c}(N)$ in $IG_{rbs}(N)$, if there is a control strategy $IG_c(N)$ in $IG(N)$. □

We have just shown that we can use the reduced interaction graph by Def. 21 for the analysis of controllability of an oWFN.

4.4 Receiving events are activated

The controller can receive messages out of the set of states the oWFN is currently in. So, in a way the controller gets information about which state the oWFN is possibly in at that time.

There are maximal states in the current node, that activate receiving events. To calculate the successor nodes we will summarize all receiving events being activated in a maximal state of the node to one single receiving event.

Example 5. Figure 6 shows the oWFN N5 (Fig. 6(a)), its complete interaction graph (Fig. 6(b)) and its reduced interaction graph (Fig. 6(c)). There are two maximal states in the root node of the interaction graph – [a, b, p4] and [c, d, p4]. The first state activates the receiving events ?a and ?b. The second state activates the receiving events ?c and ?d. For the calculation of the successor nodes we now use the receiving events ?a, ?b and ?c, ?d each as one receiving event. The edges of the reduced graph therefore are labeled with ?a,b and ?c,d (see Fig. 6(c)). ⋆

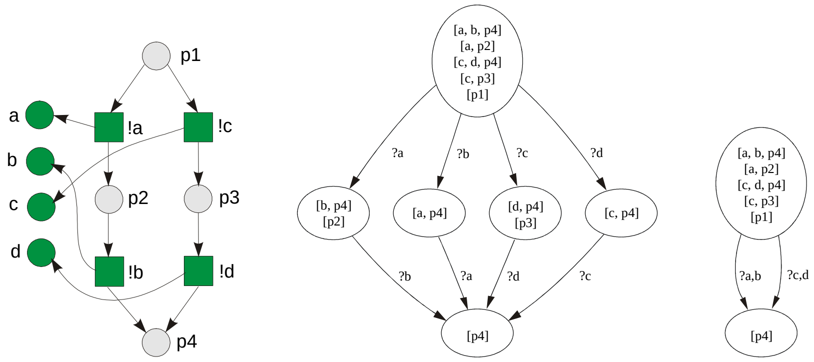
We now define the set of reduced activated receiving events of a node in the interaction graph in such a way that it fullfills the properties we just described.

Definition 22 (Summarizing Receiving Events)
Let V be a node in the interaction graph of an oWFN N. The set of the reduced activated receiving events $\mathcal{R}_{sre}(V)$[5] of a node V is defined as follows.

$$\mathcal{R}_{sre}(V) = \{\{\bigcup \mathcal{R}(z)\} \mid z \in \mathcal{Z}_{max}(V) \wedge \neg\exists z' \in (\mathcal{Z}_{max}(V) \setminus \{z\}) : \mathcal{R}(z) \supset \mathcal{R}(z')\}_{\star}$$

The set $\mathcal{R}_{sre}(V)$ now contains all receiving events activated in V. Hereby, all receiving events of one maximal state in V are summarized to one single receiving

[5] *sre* stands for "**S**ummarizing **R**eceiving **E**vents"



(a) oWFN N5 (b) Complete IG of N5 (c) Reduced IG of N5

Fig. 6. Summarizing Receiving Events

event. If there are two states z_i and z_j in V, such that the set of receiving events of z_i ($\mathcal{R}(z_i)$) is a real subset of $\mathcal{R}(z_j)$, we just add $\mathcal{R}(z_i)$ to the set $\mathcal{R}_{sre}(V)$.

Now we will define the reduced interaction graph by applying the set of receiving events $\mathcal{R}_{sre}(V)$ and the set of sending events $\mathcal{S}(\mathcal{Z}_{max}(V))$ to calculate its nodes.

Definition 23 (Reduction by Summarizing Receiving Events)
Let N be an oWFN with $\mathcal{A}(N)$ being its set of initial states. The set of the reduced activated outgoing messages $\mathcal{R}_{sre}(K)$ of node K is calculated according to Def. 22. The reduced interaction graph $IG_{sre}(N)$ of N is the directed graph $[V, E]$ with nodes V and edges E defined inductively as follows.

(i) The set of states $\mathcal{A}(N)$ is the root node of $IG_{sre}(N)$.
(ii) If $K \in IG_{sre}(N)$, K is not a terminal node by Def. 7 and there is a $M \in (\mathcal{R}_{sre}(K) \cup \mathcal{S}(\mathcal{Z}_{max}(K)))$, then K' with $K \xrightarrow{M} K'$ is a node and $[K, M, K']$ is an edge in $IG_{sre}(N)$. ⋆

We will now show, that we can use the reduced interaction graph $IG_{sre}(N)$ for the analysis of controllability of an oWFN N.

Proposition 13: Let N be an oWFN. If there is a control strategy in the interaction graph $IG(N)$, then there is a control strategy in $IG_{sre}(N)$. ⋆

Lemma 14: Letting the receiving events $E \in \{e_1, \dots, e_n\}$ occur sequentially leads to the same node as having the receiving event $E = \{e_1, \dots, e_n\}$ occur. It holds: if $V_0 \xrightarrow{e_1} V_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} V_n$, then $V_0 \xrightarrow{e_1,\dots,e_n} V_n$. ⋆

Proof (Lemma 14): We have by assumption: $V_0 \xrightarrow{e_1} V_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} V_n$ and $V_0 \xrightarrow{e_1,\dots,e_n} V'$. It is to show: $V' = V_n$.

1. $V_n \subseteq V'$. It holds by Proposition 1 $\forall i \in \{1, \dots, n\} : V_i + e_i \subseteq V_{i-1}$. Let $z \in V_n$ be chosen arbitrarily. It holds for z: $z + e_n \in V_{n-1}$. Hence, $z + e_n + e_{n-1} + \dots + e_1 \in V_0$. Let $z' = z + e_n + e_{n-1} + \dots + e_1$. z' activates the receiving events $\{e_1, \dots, e_n\}$. Let $?E = \{e_1, \dots, e_n\}$ be a receiving event. By Def. 6 we have for the set of successor states V' of V if $?E$ occurs: $V' = \{z - E \mid z \in V \land z[E] \geq 1\}$. There is a state z' in V, that complies with the necessary requirements. Therefore, we can find a state $z' - e_1 - \dots - e_n$ in V'. This state is just z.
2. $V' \subseteq V_n$. By Proposition 1 $V' + e_1 + \dots + e_n \subseteq V_0$ holds. Let $z \in V'$ be arbitrarily chosen. Hence, $z_0 = z + e_1 + \dots + e_n \in V_0$. z_0 activates the receiving events

$\{e_1, \ldots, e_n\}$. By Def. 6 we have for $i \in \{1, \ldots, n\}$: $z_i = z_{i-1} - e_i$. Therefore, $z_n = z_0 - e_1 - \cdots - e_n \in V_n$ is true. We further have $z_0 = z_n + e_1 + \cdots + e_n$ and $z = z_n$. z_n is in V_n by definition and z_n is in V' by assumption. □

Proof (Proposition 13): There is a control strategy $IG_c(N)$ in the interaction graph $IG(N)$ of the oWFN N. $IG_{sre}(N)$ is the reduced interaction graph of N by Def. 23. It is to show, that there exists a control strategy $IG_{sre_c}(N)$ in $IG_{sre}(N)$.

The root nodes of the graphs $IG(N)$, $IG_c(N)$ and $IG_{sre}(N)$ are equal by Def. 23. We set the root node of $IG_{sre_c}(N)$ equal to the root node of $IG_c(N)$. For every node with $V \in IG_c(N)$ and $V \in IG_{sre}(N)$ and V is not a terminal node by Def. 7, we add those successor nodes of V in $IG_c(N)$ to $IG_{sre_c}(N)$ that can be reached by events out of $E \in \mathcal{R}_{sre}(V)$ and $S \in \mathcal{S}_{sre}(V)$.

We claim that $IG_{sre_c}(N)$ is a control strategy in $IG_{sre}(N)$. By construction we have: the root nodes of $IG_{sre_c}(N)$, $IG_{sre}(N)$ and $IG_c(N)$ are the same. Let V be a node in $IG_{sre_c}(N)$. By construction we have $V \in IG_c(N)$. The successor node of V, being reached by events $E \in \mathcal{R}_{sre}(N)$, are in $IG_c(N)$ as well by Lemma 14 and Lemma 12. Those successor nodes, that can be reached by sending events of V in $IG_c(N)$, are in $IG_{sre_c}(N)$ by construction. If V is not a terminal node by Def. 7, there is an event for every state in V, which leads to a node in $IG_{sre_c}(N)$. If V is a terminal end node, then V is a good end node according to Def. 8 by construction.

Therefore, if there is a control strategy $IG_c(N)$ in $IG(N)$, then there is a control strategy $IG_{sre_c}(N)$ in $IG_{sre}(N)$. □

We have shown, that the reduced interaction graph by Def. 23 contains a control strategy, if the complete interaction graph has a control strategy. So, we can use this reduced interaction graph for analyzing the controllability property of an oWFN.

4.5 Combination of all Reduction Rules

In the last sections we have developed reduction rules. Then we have defined a reduced interaction graph for each rule. Furthermore, we have shown, that each reduced interaction graph can be used for the analysis of controllability of an oWFN.

In the following we will go one step further and combine the reduction rules in order to get a more compact graph, which we then can use for the analysis of controllability. As we have described already, we have integrated the first reduction rule into the other rules. That means, all rules only consider the maximal states of a node. We will now combine the reduction rules described in sections 4.2, 4.3 and 4.4.

The set $\mathcal{R}_{red}(V)$ contains the reduced activated receiving events of a node in the reduced graph.

Definition 24 (Reduced activated Receiving Events of a Node in IG_{red}) Let V be a node in the interaction graph $IG(N)$. $\mathcal{R}_{red}(V)$ is the set of the reduced activated receiving events of node V.

$$\mathcal{R}_{red}(V) = \mathcal{R}_{sre}(V)$$

The set $\mathcal{R}_{sre}(V)$ is computed according to Def. 22. ⋆

For the calculation of the set $\mathcal{R}_{red}(V)$ we use the reduction rule „Summarizing Receiving Events“. We set the set of events being calculated according to that rule to the set $\mathcal{R}_{red}(V)$.

We now want to calculate the set $\mathcal{S}_{red}(V)$. This set contains the reduced activated sending events of a node in the reduced interaction graph.

Definition 25 (Reduced activated Sending Events of a Node in IG_{red})
Let V be a node in the interaction graph $IG(N)$. $\mathcal{S}_{red}(V)$ is the set of the reduced activated sending events in node V.

$$\mathcal{S}_{red}(V) = \mathcal{S}_{nsa}(V) \cap \mathcal{S}_{rbs}(V)$$

The sets $\mathcal{S}_{nsa}(V)$ and $\mathcal{S}_{rbs}(V)$ are calculated according to Def. 17 and Def. 20. $\star$

We combine the two rules „Node- and Sending Event Analysis“ and „Receiving before Sending“. We calculate the intersection of those sending events being activated in V that (a) are not part of the black list of the node and (b) that are just activated in states of V that do not activate receiving events.

Definition 26 (Reduced Interaction Graph)
Let N be an oWFN with $\mathcal{A}(N)$ being its set of initial states. The set of reduced activated outgoing messages $\mathcal{R}_{red}(K)$ of node K is calculated according to Def. 24 and the set of reduced activated incoming messages $\mathcal{S}_{red}(K)$ of node K is calculated according to Def. 25. The reduced interaction graph $IG_{red}(N)$ of N is a directed graph $[V, E]$ with nodes V and edges E and is defined inductively as follows.

(i) The set of states $\mathcal{A}(N)$ is the root node of $IG_{red}(N)$.
(ii) If $K \in IG_{red}(N)$ and
  - K is not a bad end node by Def. 18 and K is not a terminal node by Def. 7 and there is an $M \in (\mathcal{R}_{red}(K) \cup \mathcal{S}_{red}(K))$, then K' with $K \xrightarrow{M} K'$ is a node and $[K, M, K']$ is an edge in $IG_{red}(N)$.
  - K is a bad end node by Def. 18, then no successor nodes are calculated. $\star$

We will now show that we can use the interaction graph $IG_{red}(N)$ being calculated by using all reduction rules for the analysis of controllability of an oWFN N.

Proposition 15: If there is a control strategy in the interaction graph $IG(N)$, then there is a control strategy in the reduced interaction graph $IG_{red}(N)$. $\star$

Proof (Proposition 15): Let $IG(N)$ be the interaction graph of the oWFN N with the control strategy $IG_c(N)$ according to Def. 12. Let $IG_{red}(N)$ be the reduced interaction graph of N according to Def. 26. It is to show, that there exists a control strategy $IG_{red_c}(N)$ in $IG_{red}(N)$.

The root nodes of the graphs $IG(N)$, $IG_c(N)$ and $IG_{red}(N)$ are the same by Def. 26. We set the root node of $IG_{red_c}(N)$ equal to the root node of $IG_c(N)$. For every node V with the properties $V \in IG_c(N)$ and $V \in IG_{red}(N)$ and V is not a terminal node by Def. 7, we add all those successor nodes of V from $IG_c(N)$ to $IG_{red_c}(N)$, that can be reached by events in $S \in \mathcal{S}_{red}(V)$. Further, we add those successor nodes of V from $IG_{red}(N)$ to $IG_{red_c}(N)$ that can be reached by events out of $E \in \mathcal{R}_{red}(V)$.

It is to show now that $IG_{red_c}(N)$ is a control strategy. The root nodes of $IG_{red_c}(N)$ and $IG_{red}(N)$ are the same by construction. Let V be a node in $IG_{red_c}(N)$. V is also a node in $IG_c(N)$ by construction. If V is not a terminal node by Def. 7, then there is an event for every state in V, that leads to a node in $IG_c(N)$. The successor nodes of V, that are computed by $E \in \mathcal{R}_{red}(V)$ are nodes of $IG_c(N)$ and therefore of $IG_{red_c}(N)$ by Lemma 14 and Lemma 12. The set $\mathcal{S}_{red}(V)$ consists of the activated sending events of node V that are part of $\mathcal{S}_{rbs}(V)$ as well as $\mathcal{S}_{nsa}(V)$. By Def. 20 we have that the sending events of $\mathcal{S}(V) \setminus \mathcal{S}_{rbs}(V)$ are activated by those states only that activate receiving events as well. For every such state there is by Def. 23 a receiving event in $\mathcal{R}_{red}(V)$. According to Lemma 12 all receiving events lead to nodes in $IG_c(N)$. Out of all the sending events in the set $\mathcal{S}_{rbs}(V)$ we pick those for the set $\mathcal{S}_{red}(V)$ that are not part of the black list of V. The sending events in the black list $BL(V)$ lead to bad nodes by Corollary 3.

Therefore all sending events that lead to good nodes from V are not in $BL(V)$. So, for every state in V there is at least one event, that leads to a node in $IG_c(N)$ and hence, to a node in $IG_{red_c}(N)$ as well.

Consequently, there is a control strategy $IG_{red_c}(N)$ in $IG_{red}(N)$ if there is a control strategy $IG_c(N)$ in $IG(N)$. □

We have just shown, that we can use the reduced interaction graph combining all reduction rules of the previous sections for the analysis of controllability of an oWFN.

5 Case Studies

The reduction rules presented in this paper have been implemented in Java. They were integrated into the tool *Workflow Modeling and Business Analysis Toolkit for Web Services* (WOMBAT4WS, [Mar03b,Mar03a]). WOMBAT4WS is a prototypical application of a tool kit for the analysis of web services.

With the help of the implementation of the algorithms we could test the reduction rules in practice. Table 1 shows some of the results we obtained. As input models we took processes from the BPEL specification [BIMS02] and from the PhD thesis of Martens [Mar03a]. The table shows the number of nodes (#V) and the

	oWFN		compl. IG		red. IG		reduction [%]		CG	
Process Name	#P	#T	#V	#E	#V	#E	#V	#E	#V	#E
COP	33	20	83	192	9	9	89,2	95,3	31	40
PO	30	18	183	620	7	7	96,2	98,9	34	68
eCommerce I	29	16	95	230	13	15	86,3	93,5	25	30
eCommerce II	29	16	83	192	9	9	89,2	95,3	31	40
auction service	23	14	13	19	7	7	46,2	63,2	5	4

Table 1. Sizes of the communcation graph (CG), and the complete as well as the reduced interaction graph (IG). COP stands for complex order process. The purchase order process (PO) and the auction service were taken from the BPEL specification.

number of edges (#E) of the communicaton graph (CG), of the complete interaction graph (IG), and of the reduced interaction graph for the respective model. Further it depicts the number of places (#P) and the number of transitions (#T) of the models.

As the case studies presented above show, the reduced interaction graph usually is smaller than the communication graph for the corresponding net. Besides that, we have tested our techniques using several other case studies. All showing the same result being compared to the corresponding communication graph. The reduced interaction graph is significantly smaller than the communication graph for most oWFNs. Currently, we work on finding more reduction rules in order to have $IG(N) \leq CG(N), \forall$ oWFN N measured by the number of nodes and edges of the graphs.

5.1 Benefits of the Reduction Rules

In the following we will give a brief summary of scenarios in which the reduction rules described in this paper will work best.

Maximal States. If the oWFN being analyzed consists of sending as well as of τ transitions. That means, the nodes of the interaction graph contain transient states.

Early Detection and Black List. There are sequences inside the oWFN that are not controllable. In case that there is a sending event leading to a part of the interaction graph that is not part of the control strategy, then there is a good chance that the early detection as well as the black list analysis will detect that event. Hence, the successor nodes and therefore that part of the graph will not be calculated.

Receiving before Sending. The oWFN contains sequences of sending and receiving transitions (in that order). Then we will find states that activate both kind of events. Therefore, by letting the controller receive the message first, we make sure that the successor node will not increase in size and we only calculate one successor node even though both events are activated.

Summarizing Receiving Events. There are sequences of sending transitions in the oWFN. Then we will summarize all receiving events being activated in one state to a single receiving event. We only calculate one successor node instead of several successor nodes for all receiving events activated in one state.

6 Conclusions

In this paper we presented a new possibility to analyze oWFNs – the analysis using interaction graphs. The interaction graph depicts all the possible states of an oWFN and the interaction with its controller in a very transparent way. In contrast to the communication graphs developed in [Mar03a] the nodes of the (complete) interaction graph show *all* possible states of the net. This way it is easy to understand the behavior of the net with respect to its interaction with the controller.

By analyzing the graph we have focused on *one* property of oWFNs – controllability. We have formalized how this property can be verified using interaction graphs. Therefore we defined a subgraph with certain properties, the so-called control strategy.

Since it was our goal to develop an efficient method for the controllability analysis, we have defined reduction rules for the interaction graphs. For every rule we have proven that the reduced graph can still be used for the controllability analysis. At the end we have defined an interaction graph based on the combination of all reduction rules. Our case studies show that the theoretical assumptions we have made while developing the reduction rules indeed work very well in practice.

7 Future Work

We currently adapt the results shown in this paper to a more liberal version of oWFNs. That is, as we have stated already, we want to permit final markings that do not necessarily leave the interface places empty. So far we have only considered acyclic nets. Therefore, we want to modify the techniques we have developed to fit to cyclic oWFNs as well.

Besides that, we will put more effort in finding other reduction rules. One of our goals hereby is to adapt the *partial order reduction* [Val88] to fit our needs.

References

[AIM01] Ariba, IBM, and Microsoft. *WSDL – Web Services Description Language.* W3C, Standard, version 1.1 edition, March 2001. http://www.w3.org/TR/wsdl.

[BIMS02] BEA, IBM, Microsoft, and SAP. *BPEL4WS– Business Process Execution Language for Web Services.* Vorschlag zur Standardisierung, Version 1.1, July 2002. http://ibm.com/developerworks/webservices/library/ws-bpel/.

[CST02] J.M. Colom, M. Silva, and E. Teruel. Properties. In Claude Girault and Rüdiger Valk, editors, *Petri Nets for System Engineering.*, pages 53–72. Springer-Verlag Berlin Heidelberg New York, 2002.

[Mar03a] Axel Martens. *Verteilte Geschäftsprozesse – Modellierung und Verifikation mit Hilfe von Web Services.* WiKu-Verlag für Wissenschaft und Kultur KG, 2003. Dissertation.

[Mar03b] Axel Martens. WOMBAT4WS – *Workflow Modelling and Business Analysis Toolkit for Web Services.* Institut für Informatik: Humboldt-Universität zu Berlin, 2003. `http://www.informatik.hu-berlin.de/top/wombat`.

[MRS05] Peter Massuthe, Wolfgang Reisig, and Karsten Schmidt. An operating guideline approach to the soa. Informatik-Berichte 191, Humboldt-Universität zu Berlin, 2005. Submitted to a conference.

[Sta90] Peter H. Starke. *Analyse von Petri-Netz-Modellen.* B.G. Teubner Verlag, Stuttgart, Germany, leitfäden und monographien der informatik edition, 1990.

[Val88] Antti Valmari. *State Space Generation – Efficiency and Practicality.* Dissertation, Tampere University of Technology, 1988.

[vdA98] W. M. P. van der Aalst. The application of petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[Wei04] Daniela Weinberg. Analyse der bedienbarkeit. master thesis, Humboldt-Universität zu Berlin, October 2004.